**Аннотация**

Целью дипломной работы является разработка блокчейна для проведения транзакций на основе PoA.

Дипломный проект выполнен в Microsoft Word 2021, код программы написан средствами текстового редактора PyCharm.

Работа содержит введение, n глав, заключение, библиографический список и приложение.

Данная выпускная квалификационная работа содержит: листов – n, рисунков - n, таблиц - n, формул - n, приложений - n, источников использованной литературы - n.

**Содержание**

[**Определения, обозначения и сокращения** 3](#_Toc160981955)

[**Введение** 6](#_Toc160981956)

[**Глава 1. Анализ предметной области** 10](#_Toc160981957)

[**1.1** **Изучение существующих аналогов блокчейнов** 10](#_Toc160981958)

[**1.2 Анализ потребностей и проблем, которые приложение должно решать** 11](#_Toc160981959)

[**1.3 Выбор среды разработки** 12](#_Toc160981960)

[**1.4 Изучение возможностей фреймворка FastAPI** 14](#_Toc160981961)

[**1.4.1 Преимущества фреймворка FastAPI** 15](#_Toc160981962)

[**1.4.2 Недостатки фреймворка FastAPI** 16](#_Toc160981963)

[**Глава 2. Проектирование приложения** 18](#_Toc160981964)

# **Определения, обозначения и сокращения**

В настоящей работе применены сокращения и термины с соответствующими определениями.

Блокчейн (с англ. block chain –цепь блоков)  — выстроенная по определённым правилам непрерывная последовательная цепочка блоков (связный список), содержащих какую-либо информацию. Связь между блоками обеспечивается не только нумерацией, но и тем, что каждый блок содержит свою собственную [хеш-сумму](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%81%D1%83%D0%BC%D0%BC%D0%B0) и хеш-сумму предыдущего блока.

Proof-of-Authority (PoA) (с англ. Доказательство полномочий) — это алгоритм консенсуса, подходящий для частного и разрешенного блокчейна, который опирается на ряд заранее выбранных авторитетных узлов, называемых валидаторами.

Proof-of-Stake (PoS) (с англ. Доказательство владения) — это класс механизмов консенсуса для блокчейнов, которые работают путем выбора валидаторов пропорционально их количеству активов в соответствующей криптовалюте. Это сделано для того, чтобы избежать вычислительных затрат на схемы доказательства работы (POW).

Криптоактивы — это цифровые представления стоимости, основанные на криптографии и децентрализованной одноранговой архитектуре, основанной на технологии распределенного реестра (DLT), которая позволяет двум сторонам напрямую взаимодействовать друг с другом без необходимости в доверенном посреднике.

Фреймворк (с англ. framework – «каркас, структура») – заготовка, готовая модель в IT для быстрой разработки, на основе которой можно дописать собственный код. Он задает структуру, определяет правила и предоставляет необходимый набор инструментов для создания проекта. [1]

ИС – Информационная система.

СУБД (Система управления базами данных) – это инструментальная оболочка пользователя, а ввиду того, что такая среда ориентирована на немедленное удовлетворение запросов пользователя – это всегда система интерпретатор. [2]

HTTP (протокол передачи гипертекста) – это протокол прикладного уровня для Всемирной паутины, составляющий ее самое сердце. HTTP реализуется в двух частях приложений: клиентской и серверной. Клиентская и серверная части программ, исполняемые на различных конечных системах, общаются друг с другом, обмениваясь сообщениями HTTP. Протокол HTTP определяет структуру этих сообщений и порядок обмена между клиентом и сервером. [3]

API (англ. Application Programming Interface – программный интерфейс приложения) – это набор способов и правил, по которым различные программы общаются между собой и обмениваются данными. [4]

SQL (Structured Query Language – язык структурированных запросов) – это язык программирования, который используется для работы с базами данных. [5]

PIP – система управления пакетами, которая используется для установки и управления программными пакетами, написанными на Python.

Middleware – это компонент фреймворка, который позволяет обрабатывать запросы и ответы до и после выполнения основной логики приложения.

JSON (JavaScript Object Notation) – это легковесный синтаксис для обмена данными, которые понимают как люди, так и машины. Как следует из названия, JSON основан на скриптовом языке JavaScript; однако, JSON сам по себе полностью не зависим от языка. [6]

Python – мощный компьютерный язык программирования, поддерживающий множество парадигм, который оптимизирован для обеспечения высокой продуктивности программистов, читабельности кода и качества программного обеспечения. [7]

ОС (Операционная система) – это комплекс программ, которые выступают как интерфейс между устройствами вычислительной системы и прикладными программами, предназначены для управления устройствами и вычислительными процессами, а также для эффективного распределения вычислительных ресурсов и организации надёжных вычислений. [8]

# **Введение**

Современный мир ставит перед экономикой все более сложные задачи, требующие развития инновационных подходов к финансовым операциям. В этой связи, разработка блокчейна для проведения транзакций на основе Proof of Authority (PoA) с учетом возможности децентрализации и создания независимой, но контролируемой системы становится актуальной и перспективной задачей.

Целью данной дипломной работы является исследование и разработка блокчейна, который позволит государству и крупным компаниям, обладающим доверием общества, участвовать в процессе валидации транзакций. Такой подход обеспечит высокий уровень безопасности и надежности финансовых операций, при этом сохраняя принципы децентрализации и самостоятельности системы.

Особое внимание уделяется использованию разрабатываемого блокчейна в качестве универсальной валюты при трансграничных переводах. Это позволит упростить и ускорить процесс международных финансовых транзакций, снизить их затраты и обеспечить более прозрачную и безопасную систему расчетов.

Результаты исследования данной работы могут иметь значительное значение для развития финансовой сферы, обеспечивая новые возможности для укрепления финансовой стабильности и роста экономики. Внедрение разработанного блокчейна представляет собой шаг к созданию современной и эффективной системы проведения финансовых операций в условиях быстро изменяющегося мирового рынка.

Отсутствие такого блокчейна, подходящего для пользования в России и имеющего удобный механизм валидации определяет практическую значимость данной выпускной квалификационной работы. Для решения предлагается разработать блокчейн на PoA, который обеспечит простой и надежный способ переводов криптоактивов как внутри страны, так и между странами. Пользователи будут иметь доступ к простому и понятному интерфейсу для совершения переводов, пополнений, создания подписей. За простым интерфейсом будет стоять надежная система валидации, где валидаторы будут выбраны консенсусным решением пользователей и государств.

Проанализировав существующие блокчейны на разных консенсус-механизмах и изучив запрос на методы переводов криптоактивов, можно выделить критерии, которым должен отвечать этот блокчейн.

Начало разработки блокчейн-приложения требует тщательного анализа предметной области, включая изучение существующих механизмов консенсуса и технологий блокчейн. Необходимо также провести анализ потребностей пользователей и проблем, которые блокчейн должен решать. Исследование возможностей прикладного программного обеспечения поможет определить фактические потребности, на которые должна будет отвечать разрабатываемая система. На основе полученных данных будут сформулированы требования к разрабатываемому приложению. Затем приступим к проектированию приложения, включая определение его структуры, разработку интерфейса пользователя и описание архитектуры. После этого начнем разработку приложения, включая создание базы данных, разработку логики приложения с использованием технологий Python, Docker и FastAPI, а также процесс тестирования и отладки.

Целью настоящего исследования является реализация блокчейн-приложения, ориентированного на проведение финансовых операций и транзакций с использованием принципов Proof of Authority (PoA). Основной задачей проекта является разработка удобного инструмента, способного обеспечить пользователям возможность безопасного и эффективного осуществления финансовых транзакций, а также обеспечить доступ к подробной статистической информации и аналитике.

Для разработки блокчейн-приложения на основе PoA с веб-интерфейсом и метриками для удобного проведения транзакций в России необходимо выполнить следующие задачи:

1. Изучение потребностей пользователей: Провести как можно более широкий опрос среди пользователей, чтобы понять их основные потребности и ожидания от блокчейн-приложения. Обратить внимание на их предпочтения в интерфейсе, функциональности, безопасности и удобстве использования. Учесть местные особенности и законы, которые могут повлиять на реализацию продукта.

2. Анализ существующих решений: Изучить не только успешные, но и неудачные примеры блокчейн-приложений, чтобы выявить причины их успеха или неудачи. Сравнить различные подходы и технологии, используемые в различных проектах, чтобы понять, что можно применить для достижения желаемых результатов.

3. Определить требования к приложению: Важно не только собрать список требований, но и проанализировать их приоритетность и влияние на конечный продукт. Определить основные функции и возможности для решения проблем пользователей.

4. Проектирование структуры приложения: Уделить достаточно времени для разработки детального плана архитектуры приложения. Определить основные компоненты, их взаимосвязи, обеспечить масштабируемость и гибкость системы. Учесть возможность будущего расширения функциональности и интеграции с другими системами.

5. Разработка веб-интерфейса: При создании веб-интерфейса уделить внимание не только дизайну, но и удобству использования. Провести тестирование пользовательского опыта, чтобы убедиться, что интерфейс интуитивно понятен и удобен для всех категорий пользователей. Обратить внимание на адаптивный дизайн для обеспечения корректного отображения на различных устройствах.

6. Проектирование базы данных: Разработать структуру базы данных, учитывая требования к хранению транзакционных данных и обеспечению их безопасности. Обратить внимание на методы защиты информации и резервного копирования данных, чтобы обеспечить надежность и целостность системы.

7. Реализация системы блоков и транзакций: Написать программный код, который обеспечивает создание, хранение и проверку блоков и транзакций в соответствии с выбранной концепцией PoA. Уделить особое внимание безопасности данных и процессов, чтобы исключить возможность мошенничества и сбоев в работе системы.

8. Оценка затрат и планирование развития: Провести детальный анализ затрат на разработку, внедрение и поддержку блокчейн-приложения. Разработать план долгосрочного развития продукта, учитывая не только текущие потребности пользователей, но и тенденции рынка и технологические инновации.

Выполнение перечисленных задач способствует достижению цели научной работы – разработке блокчейн-приложения для проведения транзакций на основе PoA в Российской Федерации, что приведет к созданию удобного и эффективного инструмента для пользователей, обеспечивающего интерфейс для безопасного обращения с их криптовалютными активами.

# **Глава 1. Анализ предметной области**

## **Изучение существующих аналогов блокчейнов**

Блокчейн-технология представляет собой уникальную систему, позволяющую обеспечить надежную и безопасную запись данных с использованием децентрализованной сети. Существуют различные виды блокчейнов, каждый из которых имеет свои особенности и принципы работы.

Одним из ключевых аспектов блокчейн-систем является механизм консенсуса – способ достижения единства между участниками сети относительно состояния системы. Существует несколько типов механизмов консенсуса, таких как Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS) и Proof of Authority (PoA).

Proof of Authority (PoA) является одним из наиболее эффективных типов механизмов консенсуса, основанным на доверии к участникам сети. В PoA участники, подтверждающие транзакции, должны быть идентифицированы и авторизованы, что обеспечивает высокий уровень безопасности в сети. Также PoA обладает высокой производительностью и низкими издержками, что делает его привлекательным для широкого спектра приложений.

Кроме того, блокчейн с механизмом PoA обеспечивает высокую степень гибкости при настройке параметров сети, что позволяет адаптировать его под конкретные требования проекта. Этот тип блокчейна также обладает высокой скоростью транзакций и способен обеспечить быструю и эффективную работу системы.

В результате анализа различных видов блокчейнов и их механизмов консенсуса, можно сделать вывод о том, что блокчейн с механизмом Proof of Authority (PoA) представляет собой один из наиболее перспективных и эффективных вариантов для широкого применения в различных проектах благодаря своей безопасности, производительности и гибкости.

## **1.2 Анализ потребностей и проблем, которые приложение должно решать**

Для анализа потребностей и проблем, которые блокчейн-приложение с веб-интерфейсом на алгоритме Proof of Authority (PoA) должно решать в России, необходимо учитывать несколько ключевых аспектов. Блокчейн-приложение должно обеспечивать возможность хранения истории транзакций и документов, а также гарантировать их недоступность для изменения. Блокчейн-приложение с веб-интерфейсом на PoA может ускорить процессы подписания и обмена документами, снизить издержки и упростить процедуры взаимодействия между участниками. С учетом растущей важности защиты данных и персональной информации, блокчейн-приложение должно обеспечивать высокий уровень безопасности и конфиденциальности данных. PoA алгоритм управления может помочь в обеспечении конфиденциальности данных и установлении доступа к ним на основе разрешений. Важным аспектом для блокчейна в России является соответствие законодательству и регулятивным нормам. PoA может обеспечить идентификацию участников сети, что позволит соблюдать требования к KYC (знай своего клиента) и анти-отмыванию денег. Для успешной имплементации и принятия приложения в России важно учитывать возможность интеграции с уже существующими информационными системами и сервисами, что поможет уменьшить трение при внедрении и повысить эффективность использования приложения. Анализ этих ключевых факторов поможет определить актуальность и целесообразность использования блокчейн-приложения с веб-интерфейсом.

## **1.3 Выбор среды разработки**

При выборе языка программирования без раздумий был выбран Python. Я рассмотрю причины, почему принял такое решение.

1. Простота и читаемость кода: Python известен своей простотой и понятностью синтаксиса, что делает его отличным выбором для разработки сложных систем, таких как блокчейн. Читаемый код упрощает поддержку и модификацию проекта.

2. Большое количество библиотек и фреймворков: Python обладает обширной экосистемой библиотек и фреймворков, которые могут значительно ускорить разработку блокчейн-приложений и веб-интерфейсов к ним. Например, для работы с блокчейном в Python используются библиотеки как "pyethereum", "web3.py", "pycryptodome" и многие другие.

3. Универсальность: Python поддерживается на множестве платформ и операционных систем, что обеспечивает универсальность при разработке блокчейна и веб-приложений для него. Разработчики могут создавать приложения для различных устройств и операционных систем, используя Python.

4. Поддержка многопоточности и асинхронности: Python поддерживает работу с многопоточностью, что может быть полезно при разработке высокопроизводительных приложений, таких как блокчейн. Также существует множество асинхронных библиотек для Python, позволяющих эффективно управлять асинхронными операциями.

5. Обширное сообщество разработчиков: Python имеет огромное и активное сообщество разработчиков, что означает наличие огромного количества ресурсов, обучающих материалов, форумов поддержки и др., что значительно облегчает процесс разработки и решение возможных проблем.

В совокупности эти факторы делают Python отличным выбором для разработки блокчейна и соответствующих веб-приложений, предоставляя разработчикам мощный инструментарий для создания надежных и производительных приложений.

В данном сравнительном анализе будут рассмотрены два фреймворка FastAPI и Flask, а также будут проанализированы их возможности, производительность, гибкость, расширяемость, и поддержка сообщества.

1. Производительность:

FastAPI обладает высокой производительностью за счет использования современных технологий, таких как Python 3.6+ и асинхронные запросы. Flask, в свою очередь, является более традиционным фреймворком и имеет менее эффективный механизм обработки запросов. FastAPI предлагает более быстрые скорости выполнения и обработки запросов.

1. Типизация данных и валидация:

FastAPI обеспечивает автоматическую валидацию запросов на основе типов данных, определенных в Python, что помогает предотвращать ошибки и улучшает безопасность приложения. Flask, в свою очередь, не предоставляет такой же уровень типизации и валидации данных.

1. Документация:

FastAPI автоматически генерирует документацию API на основе аннотаций Python, что делает процесс документирования API более простым и удобным. В Flask требуется дополнительное время и усилия для создания подробной документации.

1. Поддержка асинхронности:

FastAPI хорошо поддерживает асинхонное программирование и асинхронные запросы, что улучшает производительность приложения и позволяет эффективно обрабатывать большие объемы данных. Flask не так хорошо подходит для работы с асинхронными запросами.

1. Совместимость с OpenAPI и Swagger:

FastAPI построен на базе стандартов OpenAPI и Swagger, что облегчает интеграцию с другими сервисами и инструментами, поддерживающими эти стандарты. Flask не обладает такой же сильной поддержкой OpenAPI и требует дополнительных усилий для работы с ними.

В целом, FastAPI предлагает более современный подход к разработке веб-приложений, обеспечивая высокую производительность, автоматическую валидацию данных, удобную документацию и поддержку асинхронного программирования. Для блокчейна критерии асинхронности и скорости работы — это ключевые параметры для выбора фреймворка. Исходя из этого был выбран именно этот фреймворк.

## **1.4 Изучение возможностей фреймворка FastAPI**

FastAPI — это современный и эффективный веб-фреймворк на языке Python, который предоставляет разработчикам мощные инструменты для создания веб-приложений. Он позволяет быстро и легко разрабатывать API, обеспечивая высокую производительность за счет использования современных технологий.

Для разработки веб интерфейса к блокчнейну FastAPI может быть отличным выбором, так как он обладает рядом полезных возможностей. Например, благодаря асинхронной поддержке можно создать быстродействующие веб-приложение, способное обрабатывать большое количество запросов одновременно. Кроме того, FastAPI предоставляет интуитивный и простой в использовании интерфейс для описания схем данных API с помощью Pydantic, что облегчает разработку и поддержку кода.

Еще одним преимуществом FastAPI является возможность автоматической генерации документации API на основе аннотаций в коде, что значительно упрощает процесс документирования вашего приложения. Кроме того, FastAPI обладает встроенной поддержкой валидации данных, автоматической сериализации и десериализации запросов и многими другими полезными функциями.

Использование FastAPI для блокчейн-приложения позволит не только создать эффективное и надежное веб-приложение, но и изучить современные подходы к разработке API, работе с асинхронным кодом и использованию современных технологий веб-разработки.

### **1.4.1 Преимущества фреймворка FastAPI**

Одним из ключевых достоинств FastAPI является его высокая производительность. Благодаря применению современных технологий асинхронного программирования, данный фреймворк обеспечивает высокую скорость работы и эффективное использование ресурсов, что делает его отличным выбором для создания высоконагруженных веб-приложений.

Кроме того, FastAPI предлагает удобный и интуитивно понятный интерфейс для разработчиков. Он использует аннотации Python для описания эндпоинтов и схем данных API, что делает процесс создания веб-приложений более простым и эффективным. Благодаря автоматической генерации интерактивной документации на основе аннотаций в коде, разработчики могут быстро и легко создавать, и поддерживать документацию без дополнительных усилий.

Еще одним преимуществом FastAPI является встроенная поддержка валидации данных, что обеспечивает безопасность API и помогает избежать ошибок ввода. Благодаря поддержке различных форматов данных, таких как JSON, формы HTML, файлы и другие, FastAPI является универсальным инструментом для разработки разнообразных веб-приложений, обеспечивая разработчикам широкие возможности при создании функциональных и эффективных веб-сервисов.

### **1.4.2 Недостатки фреймворка FastAPI**

FastAPI, несмотря на свои многочисленные достоинства, также имеет некоторые недостатки. В частности, такой фреймворк может показаться избыточным для небольших проектов или приложений, требующих минималистического подхода. Кроме того, изучение особенностей и функционала FastAPI может потребовать времени и усилий, особенно для начинающих программистов, не знакомых с асинхронным программированием или аннотациями в Python.

Еще одним недостатком FastAPI является отсутствие широкого сообщества и большого количества сторонних библиотек и плагинов, как у более узнаваемых и распространенных фреймворков. Это может затруднить поддержку и развитие проекта в долгосрочной перспективе, особенно если возникнет необходимость в специфической функциональности, которая не предусмотрена по умолчанию в FastAPI.

Кроме того, из-за своей особенности использования асинхронного программирования FastAPI может создавать некоторые сложности в проектировании архитектуры приложения. Это может потребовать дополнительных усилий и времени для овладения соответствующими концепциями и методиками разработки, что также считается недостатком данного фреймворка.

## **1.5 Выводы по главе 1**

Анализ существующих блокчейнов и консенсус-механизмов, описание разработки блокчейна на PoA с веб-приложением на FastAPI, а также обзор фреймворка FastAPI позволяют сделать следующие выводы:

Проведенный анализ демонстрирует значимость блокчейн технологии в современном мире и ее потенциал для обеспечения прозрачности и безопасности в различных отраслях. Выбор PoA для блокчейна в условиях российских реалий обоснован его высокой производительностью и надежностью.

FastAPI, в свою очередь, представляет собой мощный инструмент для разработки веб-приложений блокчейна благодаря своей скорости работы, поддержке асинхронных запросов и автоматической генерации API документации. Однако, сложность использования FastAPI может потребовать времени на изучение.

Таким образом, сочетание блокчейна на PoA с веб-приложением на FastAPI является оптимальным выбором для разработки проектов в России, учитывая требования к производительности, надежности и удобству разработки.

# **Глава 2. Проектирование приложения**